1.   (Previously presented)   A computer-implemented method for collecting information relating to execution of an application, the method comprising:

determining a set of probe locations in the application at which collecting data relating to the execution of the application would produce non-redundant information by eliminating pairs of probe locations that would produce redundant information; and

inserting probes only at the determined probe locations in the application.

2.   (Original)   The method of claim 1, further comprising determining entry and exit points of a plurality of functions constituting at least a portion of the application.

3.   (Original)   The method of claim 2, further comprising identifying the entry and exit points as probe locations at which probes are to be inserted.

4.   (Original)   The method of claim 1, further comprising:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location.

5.   (Original)   The method of claim 4, wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

6.   (Original)   The method of claim 1, further comprising:

identifying a first location within a calling function at which execution of the application is directed to a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location.

7. (Original) The method of claim 6, wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

8. (Original) The method of claim 1, further comprising:
    identifying a block of code to which execution of the application is directed upon occurrence of an error; and
    inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

9. (Original) The method of claim 8, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

10. (Original) The method of claim 1, further comprising using the inserted probes to collect the information relating to the execution of the application.

11. (Original) The method of claim 10, further comprising analyzing the collected information.

12. (Currently amended) A computer-implemented method for collecting information relating to execution of an application, the method comprising:
    determining entry and exit points of a plurality of functions constituting at least a portion of the application;

determining a set of probe locations in the application at which collecting data relating to the execution of the application would produce non-redundant information by eliminating pairs of probe locations that would produce redundant information, the set of probe locations including at least the entry and exit points of the functions;

inserting probes only at the determined probe locations in the application;

using the inserted probes to collect the non-redundant information relating to the execution of the application; and

analyzing the collected information.

13.   (Original)   The method of claim 12, further comprising:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location,

wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

14.   (Original)   The method of claim 12, further comprising:

identifying a first location within a calling function at which execution of the application is directed to a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location,
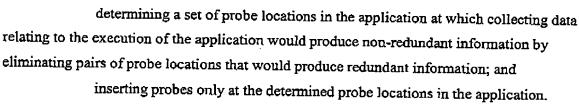
wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

Page 4 of 15

15.   (Original)   The method of claim 12, further comprising:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

16.   (Previously presented)   A computer-readable medium having an application including computer-executable instructions for:

determining a set of probe locations in the application at which collecting data relating to the execution of the application would produce non-redundant information by eliminating pairs of probe locations that would produce redundant information; and

inserting probes only at the determined probe locations in the application.

17.   (Original)   The computer-readable medium of claim 16, having further computer-executable instructions for determining entry and exit points of a plurality of functions constituting at least a portion of the application.

18.   (Original)   The computer-readable medium of claim 17, having further computer-executable instructions for identifying the entry and exit points as probe locations at which probes are to be inserted.

19.   (Original)   The computer-readable medium of claim 16, having further computer-executable instructions for:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location.

20. (Original) The computer-readable medium of claim 19, wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

21. (Original) The computer-readable medium of claim 16, having further computer-executable instructions for:

identifying a first location within a calling function at which execution of the application is directed to a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location.

22. (Original) The computer-readable medium of claim 21, wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

23. (Original) The computer-readable medium of claim 16, having further computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

24. (Original) The computer-readable medium of claim 23, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

App. No. 09/560,269
Amendment dated April 6, 2004
Reply to Office Action of January 26, 2004

25.   (Original)   The computer-readable medium of claim 16, having further computer-executable instructions for using the inserted probes to collect the information relating to the execution of the application.

26.   (Original)   The computer-readable medium of claim 25, having further computer-executable instructions for analyzing the collected information.

27.   (Currently amended)   A computer-readable medium having an application including computer-executable instructions for:

determining entry and exit points of a plurality of functions constituting at least a portion of the application;

determining a set of probe locations in the application at which collecting data relating to the execution of the application would produce non-redundant information by eliminating pairs of probe locations that would produce redundant information, the set of probe locations including at least the entry and exit points of the functions;

inserting probes only at the determined probe locations in the application;

using the inserted probes to collect the non-redundant information relating to the execution of the application; and

analyzing the collected information.

28.   (Original)   The computer-readable medium of claim 27, having further computer-executable instructions for:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location,

wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is

configured to collect the address of the second function, a second stack pointer, and a second time indicator.

29.   (Original)   The computer-readable medium of claim 27, having further computer-executable instructions for:

identifying a first location within a calling function at which execution of the application is directed to a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location,

wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

30.   (Original)   The computer-readable medium of claim 27, having further computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

31.   (Previously presented)   A computer arrangement configured to execute an application including computer-executable instructions for:

determining a set of probe locations in the application at which collecting data relating to the execution of the application would produce non-redundant information by eliminating pairs of probe locations that would produce redundant information; and

inserting probes only at the determined probe locations in the application.

32. (Original) The computer arrangement of claim 31, further configured to execute computer-executable instructions for determining entry and exit points of a plurality of functions constituting at least a portion of the application.

33. (Original) The computer arrangement of claim 32, further configured to execute computer-executable instructions for identifying the entry and exit points as probe locations at which probes are to be inserted.

34. (Original) The computer arrangement of claim 31, further configured to execute computer-executable instructions for:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location.

35. (Original) The computer arrangement of claim 34, wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

36. (Original) The computer arrangement of claim 31, further configured to execute computer-executable instructions for:

identifying a first location within a calling function at which execution of the application is directed to a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location.

37.   (Original)   The computer arrangement of claim 36, wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

38.   (Original)   The computer arrangement of claim 31, further configured to execute computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

39.   (Original)   The computer arrangement of claim 38, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

40.   (Original)   The computer arrangement of claim 31, further configured to execute computer-executable instructions for using the inserted probes to collect the information relating to the execution of the application.

41.   (Original)   The computer arrangement of claim 40, further configured to execute computer-executable instructions for analyzing the collected information.

42.   (Currently amended)   A computer arrangement configured to execute an application including computer-executable instructions for:

determining entry and exit points of a plurality of functions constituting at least a portion of the application;

determining a set of probe locations in the application at which collecting data relating to the execution of the application would produce non-redundant information by

eliminating pairs of probe locations that would produce redundant information, the set of probe locations including at least the entry and exit points of the functions;

inserting probes only at the determined probe locations in the application;

using the inserted probes to collect the <u>non-redundant</u> information relating to the execution of the application; and

analyzing the collected information.

43.   (Original)   The computer arrangement of claim 42, further configured to execute computer-executable instructions for:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location,

wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

44.   (Original)   The computer arrangement of claim 42, further configured to execute computer-executable instructions for:

identifying a first location within a calling function at which execution of the application is directed to a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location,

wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

45.   (Original)   The computer arrangement of claim 42, further configured to execute computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

46.   (New)   The method of Claim 1, wherein the determining, inserting and collecting occur at a time other than when the application is being compiled.

47.   (New)   The method of Claim 12, wherein the determining, inserting, using and analyzing occur at a time other than when the application is being compiled.

48.   (New)   The computer-readable medium of Claim 16, wherein the determining, inserting and collecting occur at a time other than when the application is being compiled.

49.   (New)   The computer-implemented medium of Claim 27, wherein the determining, inserting, using and analyzing occur at a time other than when the application is being compiled.

50.   (New)   The computer arrangement of Claim 31, wherein the determining, inserting and collecting occur at a time other than when the application is being compiled.

51.   (New)   The computer arrangement of Claim 42, wherein the determining, inserting, using and analyzing occur at a time other than when the application is being compiled.